PAT-NO: JP401102644A

DOCUMENT-IDENTIFIER: JP 01102644 A

TITLE: PIPELINE TYPE PROCESSING UNIT

PUBN-DATE: April 20, 1989

INVENTOR-INFORMATION:
NAME
SCHWARZ, ERIC M
VASSILIADIS, STAMATIS

ASSIGNEE-INFORMATION:
NAME COUNTRY
INTERNATL BUSINESS MACH CORP <IBM> N/A

APPL-NO: JP63233855

APPL-DATE: September 20, 1988

INT-CL (IPC): G06F009/38, G06F015/16

ABSTRACT:

PURPOSE: To introduce a dynamic MIMD pipeline to a computer system by reading a succeeding instruction stream even when pipeline processing and execution of a 1st instruction stream are not finished in a pipeline circuit.

CONSTITUTION: The processing unit includes receiving means 21, 22, 25 receiving an input instruction, plural pipeline processing means 26 executing an input instruction, and a dynamic activity recording table 27 to control the instruction execution of the processing means 26. Each of processing means 26a-26d stores and executes plural instructions and they are identified by definite identifiers (pipe numbers) 1-4 respectively. Since the instructions stored in each of the processing means 26a-26d are in various execution stages,

a table 27 stores information relating to the instructions stored in each of the processing means 26a-26d. The information includes an identifier of the processing means to execute the instruction. Thus, the dynamic plural-instruction stream plural-data (MIMD) pipeline is used for the computer system to attain a high efficiency.

⑲日本国特許庁（JP）　⑪特許出願公開

⑫公開特許公報（A）　平1－102644

⑤1Int.Cl.⁴	識別記号	庁内整理番号	④3公開　平成1年(1989)4月20日
G 06 F 9/38	350	A－7361－5B	
15/16	370	6745－5B	

審査請求　有　請求項の数 1　（全21頁）

⑤4発明の名称　パイプライン式処理装置

⑳特　願　昭63－233855

㉒出　願　昭63(1988)9月20日

優先権主張　㉜1987年9月30日㉝米国(US)㉛102985

⑦2発　明　者　エリック・マーク・シュワツツ　アメリカ合衆国ニューヨーク州エンデイコツト、ジエーン・レイスイ・ドライブ5エー番地

⑦2発　明　者　ステマテイズ・ヴアシリアデス　アメリカ合衆国ニューヨーク州ヴエスタル、ヴエスタル・ロード717番地

⑦1出　願　人　インターナシヨナル・ビジネス・マシーンズ・コーポレーシヨン　アメリカ合衆国10504、ニューヨーク州アーモンク（番地なし）

⑦4代　理　人　弁理士　頓宮　孝一　外1名

明　細　書

1.発明の名称　パイプライン式処理装置

2.特許請求の範囲

入力命令を受取る受取り手段と、

それぞれが一意的な識別子により識別され、複数の入力命令を保持して実行する複数のパイプライン式処理手段と、

前記処理手段及び前記受取り手段に接続され、前記複数の入力命令を保持している処理手段の識別子及び該入力命令の情報を記録して、前記処理手段での命令実行を制御するテーブル手段と、

を具備するパイプライン式処理装置。

3.発明の詳細な説明

A．産業上の利用分野

本発明は計算機システムに係り、特に、単一命令ストリーム単一データ（ＳＩＳＤ）アーキテクチヤで動作するように設計されている、浮動小数点ユニットの如き機能ユニットで、複数命令ストリーム複数データ（ＭＩＭＤ）パイプラインを使用する技術に係る。

B．従来技術とその課題

殆んどの計算機のプロセッサは何らかの形のパイプラインを利用している。パイプライン式プロセッサでは、命令ストリームの2以上の命令が同時に実行される。実行中の各命令はパイプの異なつたステージにある。パイプライン式プロセッサは当然非パイプライン式プロセッサよりも高性能である。パイプライン方式には幾つかのタイプがある。その1つは単一命令ストリーム単一データ（ＳＩＳＤ）パイプライン方式である。ＳＩＳＤタイプにおいては、個々の命令は単一のデータ・オペレーションとパイプライン結合される。しかし、ＳＩＳＤパイプライン方式を採ると多くのハザードに遭遇する。可能な最大の新データ・レートでパイプラインに入る時にハザードが生じる。ハザードは、構造ハザード及びデータ依存ハザードの2つに分けることができる。構造ハザードは、

２つのデータが同じハードウエアを使おうとした時、すなわちハードウエアの使用に関して衝突が起こつた時に生じる。データ依存ハザードは、パイプラインの１つのステージで起こつた事象がパイプラインの別のステージを介するデータ通過の可否を決定するときに生じ得る。例えば、パイプラインが２つのステージを有し、各ステージが単一メモリの使用を要求している時、一方のステージがメモリを使用中であれば、他方のステージはメモリが解放されるまで待つていなければならない。

別のタイプは、複数命令ストリーム複数データ（ＭＩＭＤ）パイプライン方式と呼ばれるものである。ＭＩＭＤタイプでは、個々の命令ではなくて命令ストリームがパイプライン化される。ＭＩＭＤタイプのパイプライン方式はハザードの問題を生じない。しかし、命令ストリームがパイプライン化されると云つても、或る命令ストリームの実行が完了しない限り、次の命令ストリームの実行を開始することはできない。従つて、ＭＩＭＤパイプライン方式の性能は、ＳＩＳＤパイプライン方式よりは上であるが、一時に１つの命令ストリームしか実行できないと云う原理によつて制限を受ける。

第２図を参照しながら、従来のＭＩＭＤパイプライン方式について説明する。

第２図において、記憶装置１０は複数の命令ストリーム及び各命令ストリームの状態を記憶している。記憶装置１０の出力には初期設定制御部１２が接続され、そして初期設定制御部１２の出力にはパイプライン回路１４が接続されている。パイプライン回路１４はハザード検出回路を持つていない。パイプライン回路１４の出力は記憶装置１０に接続されている。

動作時には、まず１つの命令ストリームが記憶装置１０から初期設定制御部１２へ送られる。初期設定制御部１２は、受取つた命令ストリームの各命令を１つずつパイプライン回路１４へ送る。これらの命令はパイプライン回路１４の内部でパイプライン化され、一時に１つずつ実行される。

実行後、更新された命令はパイプライン回路１４から記憶装置１０に書込まれる。命令ストリームの最後の命令が初期設定制御部１２からパイプライン回路１４へ送られ、その実行が終つて最後の更新された命令が記憶装置１０へ送られると、別の命令ストリームがパイプライン回路１４での実行のために、記憶装置１０から初期設定制御部１２へ読出される。

上述から明らかなように、第２図の構成においては、最初の命令ストリームのパイプライン化及び実行がパイプライン回路１４で終らない限り、次の命令ストリームを記憶装置１０から初期設定制御部１２へ読出すことはできない。これが従来のＭＩＭＤパイプライン方式の欠点である。

従つて本発明の目的は、上述の如き欠点のない新しいタイプのパイプライン（以下、動的ＭＩＭＤパイプラインと云う）を計算機システムに導入することにある。

C．課題を解決するための手段

本発明に従うパイプライン式処理装置は、入力命令を受取る受取り手段、入力命令を実行する複数のパイプライン式処理手段、及び処理手段での命令実行を制御するためのテーブル手段を含む。各処理手段（パイプ）は複数の命令を保持して実行することができ、それぞれ一意的な識別子（パイプ番号）により識別される。各処理手段に保持されている命令は様々な実行段階にあるため、処理手段における各命令の状況を正確に記録しておくことが必要である。そのため、テーブル手段（動的活動記録テーブル）は各処理手段に保持されている命令に関する情報を記憶する。この情報には、当該命令を実行する処理手段の識別子も含まれる。

上述の如きパイプライン式処理装置は、例えば浮動小数点ユニットに適用できる。一般に、浮動小数点ユニットはＣＰＵの如き他の機能ユニットからの要求に応答して動作するが、命令実行で手

一杯の時は要求に応じられないことがある。また、何らかのハザードが存在していると、命令の実行を遅らせる必要がある。これらを監視するため、後述の実施例では、ハンドシェーク／大域ハザード回路が使用される。ユニットが使用中でなく且つハザードが存在しなければ、次の命令を対応する処理手段へ入れることができる。また、命令の長さや種類に応じてＭＩＭＤモード又はＳＩＳＤモードを選択するＭＩＭＤ／ＳＩＳＤ切替え回路も設けられる。

D．実施例

以下では、本発明に従う動的ＭＩＭＤパイプラインを計算機システムの機能ユニットに組込んだ例を説明する。機能ユニットとしては浮動小数点ユニット（以下、ＦＰＵと略称）を取り上げるが、勿論本発明はこれに限定されるものではない。

計算機システムは、ＦＰＵの他に、キャッシュ、ＣＰＵ、及びベクトル・プロセッサを含む。ＦＰＵは、キャッシュ、ＣＰＵ又はベクトル・プロセッサからデータを直接受取る。命令はＣＰＵから受取る。ＣＰＵは、キャッシュから来るデータの制御には関与しない。ＣＰＵは、（キャッシュから）データを要求する一方で、命令をＦＰＵへ送る。データがキャッシュからアクセスされている間、ＣＰＵは命令をＦＰＵへ送り続ける。その際、キャッシュからデータをアクセスするサイクルと、対応する命令をＦＰＵへ送るサイクルとの間の同期は無視される。従つて、例えばサイクルＮでＦＰＵに到着したデータが、サイクルＭ（ただしＭ≦Ｎ）でＦＰＵに送られた命令に関係するデータである場合がある。ＣＰＵは、Ｃバスと呼ばれるバスを介して、ＦＰＵ及び他のユニット（例えばキャッシュ）に動作を要求する。Ｃバスは、ＣＰＵとＦＰＵとの間で命令を転送する唯一の手段である。命令のＯＰコードの他に、ハンドシェーク制御信号もＣバス上を転送される。ＣＰＵが要求を送る機能ユニットをプロセッサ・バス・ユニット（ＰＢＵ）と呼ぶ。その１つがＦＰＵである。ＣＰＵが自身で実行できない命令を検出し、ＰＢＵがその命令を実行すべき場合には、ＣＰＵは適当なＰＢＵにプロセッサ・バス動作（ＰＢＯ）信号を送る。例えば、ＣＰＵで解読した命令が長精度形式の浮動小数点乗算命令であれば、ＦＰＵで実行した方が適しているので、ＣＰＵはＰＢＯ信号をＦＰＵへ送つて、この命令の実行を要求する。

ＦＰＵは２つの主要部、すなわちデータが実際に流れる部分と、命令が導入されて制御信号に変換される部分とを含んでいるが、本明細書で主に説明するのは後者の部分である。

本発明の動的ＭＩＭＤパイプラインを利用するＦＰＵの構成例を第１図に示す。

第１図において、ＦＰＵ２０のＣバスは命令スタック２１に接続される。命令スタック２１の出力はデコーダ２２に接続される。デコーダ２２の出力はハンドシェーク／大域ハザード回路２３、ＭＩＭＤ／ＳＩＳＤ切替え回路２４、及び初期設定回路２５に接続される。回路２３及び２４の出力は初期設定回路２５の入力に接続される。初期設定回路２５の出力は動的活動記録テーブル２７、パイプライン機構２６、及び複数の浮動小数点レジスタ（ＦＰＲ）のアレイ２８に接続される。ハンドシェーク／大域ハザード回路２３の出力は例外処理回路１９にも接続される。例外処理回路１９の出力は動的活動記録テーブル２７に接続される。パイプライン機構２６も動的活動記録テーブル２７及び例外処理回路１９に接続され、またＤバスに出力を発生する。Ｄバスは、データ・キャッシュ（図示せず）及びアーキテクチャ上で定義されたローカル・メモリであるＦＰＲアレイ２８に接続される。動的活動記録テーブル２７の出力は、Ｄバス及びＦＰＲアレイ２８への出力ゲート動作を制御するのに用いられる。Ｃバスは、命令スタック２１の他に、Ｄバス・スタック制御回路３０にも入力を供給する。Ｄバス・スタック制御回路３０の出力はＤバス・スタック２９に接続される。Ｄバス・スタック２９はＤバスから入力を受取る。Ｄバス・スタック２９及びＦＰＲアレイ２８の出力は、データ・フローを開始するデータを発生する。

本発明の動的ＭＩＭＤパイプラインは２つの経路、すなわち命令及び制御のための経路（Ｃバス経路）と、データ・フローのための経路（Ｄバス経路）に分けることができる。命令はＣバスを介して受取られ、命令スタック２１に蓄かれた後、デコーダ２２で解読される。データはＤバスを介して第１図の動的ＭＩＭＤパイプラインへ導入される。

ハンドシエーク／大域ハザード回路２３はＣＰＵ（図示せず）へハンドシエーク信号を送り、大域ハザードを検出する。ハンドシエーク／大域ハザード回路２３の詳細は第１０図に示してあるが、これについてはあとで説明する。Ｃバスは、ＣＰＵと各ＰＢＵ（ＦＰＵ２０を含む）との間で一組のハンドシエーク信号を転送する。ＦＰＵ２０がＣバスを介して要求を受取つた場合、その要求がＣＰＵからのもので、且つＦＰＵ２０が当該要求に関係する唯一のＰＢＵであれば、ハンドシエーク／大域ハザード回路２３は、肯定応答信号（ＡＣＫ）、使用中信号又は割込み信号をＣＰＵへ返送する必要がある。ＡＣＫは、Ｃバスから要求を受取つた時にＦＰＵが使用中（ビジー）でなければ、ＦＰＵからＣＰＵへ送られる。割込み信号は、データ例外が生じて状況ワードに重要情報が書込まれていると、ＦＰＵからＣＰＵへ送られる。使用中信号は、ＦＰＵが別の命令を受入れて実行することができない場合に、ＦＰＵからＣＰＵへ送られる。

ハンドシエーク／大域ハザード回路２３は、大域ハザードを検出すると、その存在を表わす信号を初期設定回路２５へ送る。また回路２３は、（初期設定回路２５と関連して）ＦＰＵ２０の応答を他のプロセッサ・バス・ユニット（ＰＢＵ）へ送る。回路２３は、命令ストリームの始め及び終りを検出するのにも役立つ。回路２３は、実行中の他の命令に対する命令のデータ依存性によるハザード（データ・インターロック）の存在を検出する。

ＭＩＭＤ／ＳＩＳＤ切替え回路２４は、デコーダ２２で解読された入力命令に応じて、ＳＩＳＤモード又はＭＩＭＤモードを設定する。入力命令が６４ビットよりも長いオペランドを使用するものであるか、又は命令の実行が困難であれば、ＭＩＭＤ／ＳＩＳＤ切替え回路２４はＳＩＳＤモードを選択し、さもなければＭＩＭＤモードを選択する。

実行が困難であると考えられ、ＳＩＳＤモードが選択される命令は次の通りである。

浮動小数点除算
固定小数点除算
平方根演算
拡張精度オペランドを用いる演算

ＳＩＳＤモードでの実行中は、当該命令の実行を除くと、どのような活動も行われない。これは、ＦＰＵ２０のハンドシエーク回路２３からＣＰＵへ送られる使用中信号を活動状態に保つことにより達成される。使用中信号が活動状態にあると、ＣＰＵは新しい要求をＦＰＵ２０へ送れない。

以下に列挙する命令又はその如何なる組合せも、ＭＩＭＤ／ＳＩＳＤ切替え回路２４にＭＩＭＤモードを選択させる。

浮動小数点演算
　加算
　比較
　二分
　ロード
　乗算
　記憶
　減算
固定小数点演算－マイクロコード
　乗算
その他の演算－マイクロコード
　ロード
　記憶
　状況ワード
　間接モード
　再試行

次の命令は、ＭＩＭＤ／ＳＩＳＤ切替え回路２４にＳＩＳＤモードを選択させる。

浮動小数点－マイクロコード

拡張精度加算
拡張精度乗算
除算
拡張精度除算
平方根演算
拡張精度丸めロード
固定小数点演算ーマイクロコード
除算

初期設定回路２５はパイプを起動し、動的活動記録テーブル２７を更新する。初期設定回路２５の詳細については、あとで第１１図を参照しながら説明する。初期設定回路２５はハンドシエーク／大域ハザード回路２３と関連して、命令ストリームの始め及び終りを決定し、また何らかのデータ依存ハザードが存在するかどうかを決定する。命令解読後、デコーダ２２からの出力により示される命令のタイプが、初期設定回路２５及びハンドシエーク／大域ハザード回路２３において、使用する適切なパイプの第１サイクル状況の完了状況（内部パイプ制御部２６ａ～２６ｄにより示される）と比較される。大域ハザードがないことが動的活動記録テーブル２７により示され、当面の内部ハザードがないことがハンドシエーク／大域ハザード回路２３の出力により示されると、命令の初期設定が行われる。ハンドシエーク／大域ハザード回路２３が使用中信号を発生していると、初期設定回路２５は如何なる初期設定も行わない。

初期設定は、適切なパイプの状況制御の開始、及び動的活動記録テーブル２７への新しいエントリの入力を含む。初期設定の状況をＣＰＵに知らせるのはハンドシエーク／大域ハザード回路２３である。回路２３は、ＡＣＫをＣＰＵへ送ることにより、命令の処理が開始されたことを示し、また使用中信号をＣＰＵへ送ることにより、ＦＰＵ２０が命令を受取つたが、それ程多くの命令を処理できないため、入力命令のパイプを停止することを示す。前述のように、回路２３及び２５は命令ストリームの始め及び終りを決定する。まだ命令ストリームにない命令に対するＡＣＫ及び使用可（使用中でない）は命令ストリームの始めを示し、使用中信号での応答は命令ストリームの終りを示す。ハンドシエーク／大域ハザード回路２３は、データ依存性によるハザードの存在を検出するのに用いられる。初期設定回路２５は、動的活動記録テーブル２７に新しいエントリを追加する。従つて、要約すると、初期設定は、ハンドシエーキング、活動記録ファイルの更新、及びもしデータ依存ハザードが生じていればその処理、から成る。

第１図の動的ＭＩＭＤパイプラインは４つのパイプライン回路（パイプ１～パイプ４）２６ａ～２６ｄを含む。従つて命令のカテゴリも４つある（１つのパイプ当り１つのカテゴリ）。

Ｄバス上のデータはＦＰＲアレイ２８又はＤバス・スタック２９により処理される。Ｄバス・スタック２９はＤバス・スタック制御回路３０の制御を受ける。

例外処理回路１９は、例外が生じているかどうかを調べる。命令実行中に生じ得るデータ例外のタイプは次の通りである。

指数オーバーフロー例外
指数アンダーフロー例外
浮動小数点除算例外
固定小数点除算例外
有効数字例外
平方根演算例外

これらの例外のうちの１つを起こす命令が検出されると、この命令の後で受取られたすべての命令は、例えそれらが既に実行中であつたとしても、あたかも受取られなかつたかの如き効果を与えるため、取消さなければならない。これは、動的ＭＩＭＤアーキテクチヤが維持しなければならないＳＩＳＤアーキテクチヤの性質である。命令の取消しは、例外を起こした命令の完了後に動的活動記録テーブル２７の中のすべての有効ビットをゼロに変えることにより為される。更に、ＣＰＵ及び他のユニットは割込みを知らされ、ＣＰＵが割込み処理ルーチンを開始するまでそれらの命令を取消さなければならない。

計算機システムのＦＰＵに配置される第１図の

動的ＭＩＭＤパイプラインはＣバスから命令を受取り、ＦＰＵは他のＰＢＵと同様に、「ＡＣＫ」、「使用中」及び「割込み」の如き特定のハンドシエーク信号を送ることによつて応答する。ＣＰＵはパイプライン・モードで働き、サイクル毎にＰＢＯコマンドを、ＡＣＫが返されたかどうかには無関係に送出するので、ＰＢＵは次のＰＢＯの実行に進む前に、最後のＰＢＯに対するＡＣＫが返されたかどうかを調べなければならない。ＰＢＵはスマート・インタフエースを含んでおり、それを用いて、他のＰＢＵとＣＰＵとの間のハンドシエーク状態をチエックする。ＰＢＵは、ＰＢＯを受取つた後のサイクルで、３つのハンドシエーク信号のうちの１つを回路２３からＣＰＵに送る必要がある。ＰＢＵは、ハザードに遭遇すると、使用中信号をＣＰＵに送る。その時ＰＢＵは、受取つた命令及びその次の命令を命令スタック２１に保持しており、かくしてＣＰＵからの命令の順序が維持される。

第３図の(a)に示すように、命令スタック２１はＣバス・スタック２１ａ及びＣバス・レジスタ２１ｂを含む。受取つた命令はＣバス・レジスタ２１ｂに保持され、その次の命令はＣバス・スタック２１ａに保持される。使用中信号を発生させるハザードに遭遇しない限り、命令はスタックされない。ＦＰＵ２０は、処理能力の範囲内であれば、どれ程多くの命令でも受入れるが、ＣＰＵ程多くの情報を含まない。と云うのは、ＣＰＵはその命令バツフアの内容から、何らかの問題が生じそうだと判断すると、命令をバス・ユニットへ送る前にその命令を停止させることができるからである。ＦＰＵ２０及び他のバス・ユニット（例えばデータ・キヤツシユ）での実行を要求するＰＢＯがＣＰＵから送られる場合、ＦＰＵ２０はデータ・キヤツシユによる命令実行の開始を阻止できない。従つて、ＦＰＵ２０によるパイプライン化の最も効率的な方法は、ハザードに遭遇するまでにできるだけ多くの処理を行うことである。

第３図において、(a)は命令スタック２１の構成を示し、(b)は配線モード時におけるＣバス上のビットを示し、(c)はマイクロコード・モード時におけるＣバス上のビットを示している。

前述のように、命令スタック２１はＣバス・スタック２１ａ及びその出力に接続されたＣバス・レジスタ２１ｂを含む。Ｃバスと同じく、命令スタック２１は多くとも２つの命令につき２５ビットの情報を保持する。各ビットの意味は次の通りである。

ビット０－ＦＰＵ２０が配線モードにあるかマイクロコード・モードにあるかを示すＰＢＯビット。配線モード（ビット０＝０）においては、例外はＣＰＵに報告される。マイクロコード・モード（ビット０＝１）においては、例外は状況ワード（第８図参照）に記憶されるが、報告されることはない。

ビット１（ＦＰ）－当該命令がＦＰＵ２０で実行されねばならないことをＦＰＵ２０に知らせるＦＰＵ要求ビット。

ビット２（ＩＰＵ）－命令の解読をキヤツシユに知らせるＩＰＵ／キヤツシユ要求ビット。

ビット３（ＶＰ）－ベクトル・プロセツサ要求ビット。

ビット４～１０－命令のＯＰコード。

ビット１１～１３（Ｆ１）－オペランド１の符号化されたＦＰＲアドレス。

ビット１４～１６（Ｆ２）－オペランド２の符号化されたＦＰＲアドレス。

ビット１７～１９（ＴＡＧ又はＳＲＣ）－配線モードにおいては、これらのビットは、例外発生に伴つて状況ワードに記憶される割込みタグ（ＴＡＧ）を表わし、マイクロコード・モードにおいては、ソースＰＢＵ（ＳＲＣ）を識別する。割込みタグは、ＣＰＵの命令スタックにある命令を一意的に識別する。

ビット２０～２２（ＤＳＴ）－マイクロコード・モードにおいて宛先ＰＢＵを識別する。

ビット２４（Ｐ）－命令の妥当性を検査するためのパリテイ・ビット。

Ｃバス上の命令は上述の２５ビットの形で命令スタック２１へ導入される。なお、第３図の(b)

及び（c）の中の斜線部分は予約フィールドを表わしている。

動的活動記録テーブル２７の構成を第４図に示す。第４図の例では、動的活動記録テーブル２７は、最大８個の命令までそれぞれ１７ビットの情報を記憶する。これは、入力命令が４つのパイプライン回路２６ａ～２６ｄのうちの１つに入つて完了する必要があるときに使用される。命令はスタックされるので、テーブル２７は１以上の命令ストリームの命令実行の完了を順序づける手段を提供する。Ｃバスは一時に１つの命令しか送れないので、この結果として命令の開始時間が決まる。１以上の命令ストリームの命令の実行は完了までに複数のサイクルを要することがあり、またパイプも複数存在しているので、複数の命令を同時に実行することが可能である。アーキテクチャ上の制約から、割込みが起こつた時に命令がシーケンシャルでなかつたなら、結果が予測できないことがあるので、命令完了の順序は維持されねばならない。従つて、順序づけ情報及び完了情報をテーブル２７に保持しておく必要がある。テーブル２７は次のような情報を記憶する。

ビット０（Ｖ）－有効ビット。

ビット１～３（ＷＲ　ＡＤＤＲ）－書込みアドレス。

ビット４～５（ＰＩＰＥ　ＮＯ）－パイプ番号（００＝加算、０１＝乗算、１０＝ロードＲＸ、１１＝その他）。

ビット６（Ｈ）－１であれば配線ＦＰＵ要求であることを示し、０であればマイクロコード命令であることを示す。

ビット７（ＷＴ）－書込みタイプの命令かどうかを示す。

ビット８（Ｍ／Ｓ）－１であればＭＩＭＤ命令タイプであることを示し、０であればＳＩＳＤ命令タイプであることを示す。

ビット９（ＥＸＴ）－拡張精度結果を書込むかどうかを示す。

ビット１０（ＬＥＮ）－１であれば結果が長精度であることを示し、０であれば短精度であることを示す。

ビット１１～１３（ＩＮＴ　ＴＡＧ）－ＣＰＵのスタックにおける命令を一意的に識別する割込みタグ。

ビット１４～１６（ＰＳＷ　ＰＴＲ）－命令の再試行ポインタを識別する。

複数のパイプライン回路２６ａ～２６ｄを順序づけるという点でパイプ番号（ＰＩＰＥ　ＮＯ）は重要である。パイプが１つだけであれば、順序づけは殆んど問題にならないが、複数パイプのシステムでは、追跡情報を維持しておかねばならない。書込みアドレス（ＷＲ　ＡＤＤＲ）、書込みタイプ（ＷＴ）及び結果長（ＬＥＮ）は命令を完了させるのに有用である。割込みタグ（ＩＮＴ　ＴＡＧ）は、例外が生じた時に記憶され、この例外を起こした命令を識別する。もしそれがＳＩＳＤ命令であれば、パイプの終りに有効データがあるかどうかを見る代りに、サイクルを計数するカウンタによつて命令の完了が感知される。最も重要なビットは、対応する命令が有効かどうかを示す有効ビット（Ｖ）である。有効ビット（Ｖ）は例外が生じるとクリアされる。有効ビット（Ｖ）は命令完了時にもクリアされるが、その場合はスタックが上方にシフトされる。従つて、動的活動記録テーブル２７において有効ビット（Ｖ）をクリアすることにより、ＦＰＵ２０中のすべての未了命令を迅速に取消すことができる。

パイプライン機構２６の構成を第５～８図に示す。第５図は、加算、減算、除算、比較、平方根演算等の加算タイプの命令に用いられる加算パイプを含むパイプライン回路２６ａの構成を示している。第５ａ図の加算パイプは３サイクルでその機能を遂行する。第６ａ図は乗算命令に用いられる乗算パイプを含むパイプライン回路２６ｂの構成を示しており、これは５サイクルでその機能を遂行する。第７図はＲＸタイプのロード命令に用いられるパイプライン回路２６ｃの構成を示しており、これは２サイクルでその機能を遂行する。第８図は他のすべての機能（普通は補助レジスタや状況レジスタの書込み又は読取り）を遂行する

ためのパイプライン回路２６ｄの構成を示している。

パイプライン回路２６ａ～２６ｄはそれぞれ制御部及びパイプ部（１～４）を含んでいる（第１図参照）。制御部は、関連するパイプ中の流れをできるだけ遠くまで行かせ且つＦＰＲアレイ２８がインターロックされる時を感知することにより、及びどこに良好なデータがあるかを調べることにより、関連するパイプの内部を制御する。ＭＩＭＤモードでは、パイプ１～４が異なつた長さを持つているため、これらのパイプの全体的な制御が複雑になる。

パイプ内部には関連する状況フイールドを有する幾つかのレジスタがある。第５ａ図の加算パイプの場合、第５ｂ図に示した状況フイールドの各ビットの意味は次の通りである。

ビット０～２（ＡＤＤＲ）－オペランドのＦＰＲアドレス。

ビット３（ＶＩ）－パイプ中の当該ステージにおける命令が有効かどうかを示す。

ビット４（ＶＤ）－関連するレジスタ中のデータが有効かどうかを示す。

ビット５（Ｍ／Ｓ）－ＭＩＭＤモードかＳＩＳＤモードかを示す。

ビット６（ＲＸ）－命令がＲＸタイプかＲＲタイプかを示す。

ビット７（２ＢＹ）－これが１になつていると、２サイクルのバイパスが行われていることを示す。

第６ａ図の乗算パイプの場合は、更に次のような４ビットの状況情報を含む。

ビット８（ＥＸＴ）－拡張精度結果かどうかを示す。

ビット９（ＬＩ）－オペランドが長いかどうかを示す。

ビット１０（ＦＬＰ）－１であれば浮動小数点乗算であることを示し、０であれば固定小数点乗算であることを示す。

ビット１１（ＩＮＴＬ）－Ｙのオペランドがインターロックされるかどうかを示す。

にお、乗算パイプにおいては、状況フイールドのビットＳ（ＶＲ）は、パイプの当該ステージが有効な命令を持つていなくてもデータ（結果）が有効であるかどうかを示す。他のパイプ３及び４は短いので、状況情報は不要である。

パイプの各ステージに関する状況情報は、次のステージにその有効性を知らせるものであり、もし問題がなければ、次のステージは次のサイクルで有効になる。このように、状況情報は、問題の有無を判断したり、命令のデータをパイプ中のできるだけ遠くまで流すようにしたりするのに有用である。当該パイプによるＤバスへのデータ出力及び実行後に、当該パイプは、そのパイプ番号が動的活動記録テーブル２７の最も古いエントリ中のパイプ番号（ＰＩＰＥ　ＮＯ）と一致するのを待つ。一致した時、当該パイプは完了を許され、かくして命令完了の同期が維持される。

第５ａ図に示した加算パイプ（パイプ１）は、整列レジスタ３４、ＦＡレジスタ３１、ＦＢレジスタ３２、Ａレジスタ３５、Ｂレジスタ３６、加算器３７、ＦＳレジスタ３３、Ｓレジスタ３８、及び事後正規化レジスタ３９を含んでいる。第５ｂ図に示した状況フイールドは、ＦＡレジスタ３１、ＦＢレジスタ３２及びＦＳレジスタ３３に関連している。

第５ａ図の加算パイプは３サイクルでその機能を遂行する。第１サイクルでは、データがＦＰＲアレイ２８又はＤバスから検索され、整列レジスタ３４で整列操作が行われ、そしてオペランドがＡレジスタ３５及びＢレジスタ３６にラッチされる。第２サイクルでは、加算器３７で実際の加算が行われ、その結果がＳレジスタ３８に書込まれる。最後の第３サイクルでは、事後正規化レジスタ３９が必要に応じて先行ゼロをシフトアウトし、データをＦＰＲアレイ２８に戻す。

上述の機能は、第５ａ図のパイプが加算命令を処理するときのものであるが、加算タイプに属する他の命令の場合は、内部バイパス制御のために他のレジスタも使用される。複数の異なつた命令を処理し得る第５ａ図のパイプを維持し制御するためには、３つの主制御レジスタが必要である。

それらは、ＦＡレジスタ３１、ＦＢレジスタ３２及びＦＳレジスタ３３である。

前述のように、レジスタ３１、３２及び３３の状況フィールドは以下のビットを含んでいる。

1. インターロックされ得るオペランドを見つけるのに用いられるＦＰＲアドレス・ビット（ＡＤＤＲ）。

2. パイプ中の当該ステージが命令に対して有効であることを示すのに用いられる有効命令ビット（ＶＩ）。

3. 関連するデータ・レジスタが有効であることを示す有効データ・ビット（ＶＤ）。

4. 命令の終りを知らせるＭＩＭＤ／ＳＩＳＤパイプ標識（Ｍ／Ｓ）。ＭＩＭＤモードの場合は、最終ステージが有効で、完了について競合がなければ、命令の終りである。ＳＩＳＤモードの場合は、命令がパイプ中を何回かループすることがあるので、もう少し複雑になる。

5. 命令がＲＸタイプであることを示すビット（ＲＸ）。これは、ＦＢレジスタ３２に関しては、そのアドレス・ビットが無効で、入力データ（まだ有効でなければ）のためにデータ・バスを監視する必要があることを示す。

6. ２サイクル・バイパスの第１サイクルを示すビット（２ＢＹ）。インターロックされたデータが見つかつた場合、それを取出すのに２サイクルを要することがある。

第６ａ図の乗算パイプ（パイプ２）は、ＦＸＡレジスタ４１、ＦＹＳレジスタ４２、ＦＸＢレジスタ４３、ＦＹレジスタ４４、ＦＰレジスタ４５、ＸＡレジスタ４６、３Ｘハードウエア４７、ＸＢ／３Ｘレジスタ４８、Ｙレジスタ４９、Ｍ１ハードウエア５０、Ｍ２ハードウエア５１、及びＰレジスタ５２を含んでいる。この乗算パイプは、もしハザードに遭遇しなければ、５サイクルでその機能を遂行する。

サイクル１－オペランド１がＦＰＲアレイ２８からＸＡレジスタ４６にロードされる。バス幅の制限から一時に１つのオペランドしかロードできないので、もしオペランド２もＦＰＲアレイ２８からロードするのであれば、オペランド２はＦＰＲアレイ２８から読出されて一時レジスタに記憶される。

サイクル２－オペランド２が一時レジスタ又はＤバスからＹレジスタ４９にロードされ、同時にオペランド１の３倍乗算が３Ｘハードウエア４７で行われ、結果がＸＢ／３Ｘレジスタ４８の３Ｘ部に書込まれる。更に、ＸＡレジスタ４６の内容が直接ＸＢ／３Ｘレジスタ４８のＸＢ部にロードされる。

サイクル３及びサイクル４－これらは乗算器の実際の実行サイクルで、Ｍ１サイクル及びＭ２サイクルと呼ばれ、Ｍ１ハードウエア５０及びＭ２ハードウエア５１を使用する。Ｍ１サイクルと Ｍ２サイクルの間にレジスタが介在することはなく、従つてＸＢ／３Ｘレジスタ４８及びＹレジスタ４９は、データがＰレジスタ５２にラッチされるまで、これら２サイクルの間保持されねばならない。

サイクル５－Ｐレジスタ５２からＦＰＲアレイ２８への書込みが行われる。本実施例では、チップ間に１本の８バイト・データ・バスしか設けていないので、拡張精度結果の場合には、第２の書込みサイクルであるサイクル６が後に続く。

乗算パイプを制御する制御レジスタは、ＸＡレジスタ４６の状況を維持するＦＸＡレジスタ４１、ＲＲタイプの命令の場合にオペランド２を最初に受取る一時レジスタの状況を維持するＦＹＳレジスタ４２、ＸＢ／３Ｘレジスタ４８の状況を維持するＦＸＢレジスタ４３、Ｙレジスタ４９の状況を維持するＦＹレジスタ４４、及びＰレジスタ５２の状況を維持するＦＰレジスタ４５である。これらの制御レジスタないし状況レジスタは、第６ｂ図に示す１２ビットの情報を維持する。これらのビットは次の通りである。

1. インターロックされ得るオペランドを見つけるのに用いられるＦＰＲアドレス・ビット（ＡＤＤＲ）。

2. パイプ中の当該ステージが命令に対して有

効であることを示すのに用いられる有効命令ビット（ＶＩ）。

3. 関連するデータ・レジスタが有効であることを示す有効データ・ビット（ＶＤ）。

4. 別のチップ上でローカル作業用記憶域を作り出すのに用いられる有効結果ビット（ＶＲ）。ＲＲ命令の場合、オペランド2は、別の命令がその内容を変更するまでは、乗算後もＹレジスタ４９で有効になつている。有効結果ビットは、Ｙレジスタ４９にあるデータが定義されたアドレスに関して有効であることを示す。また、ＲＲ命令及びＲＸ命令の場合、乗算の結果を含むＰレジスタ５２は、別の乗算がこのパイプで行われるか、又は別の命令がオペランド１のＦＰＲアドレスを変更するまでは、オペランド１によつてアドレス指定されるＦＰＲと同じである。これは性能を上げる上で極めて重要である。ロードを少しでも減らすことができれば、性能の向上につながる。

5. 命令がＲＸタイプで、そのアドレス・ビットが無効であることを示すビット（ＲＸ）。入力データがまだ有効でなければ、そのためにデータ・バスを監視する必要がある。

6. ２サイクル・バイパスの第１サイクルを示すビット（２ＢＹ）。インターロックされたデータが見つかつた場合、それを取出すのに２サイクルを要することがある。

7. 拡張精度結果をＦＰＲアレイ２８に書戻さなければならないことを示すビット（ＥＸＴ）。

8. 長いオペランドがレジスタにあることを示すビット（ＬＩ）。

9. パイプ中の当該ステージにある命令が浮動小数点命令であることを示すビット（ＦＬＰ）。

10. Ｙレジスタ４９のオペランドに対するインターロックを示すビット（ＩＮＴＬ）。

ＦＸＡレジスタ４１及びＦＹＳレジスタ４２は、サイクル１で初期設定回路２５によりセットされる。ＸＢ／３Ｘレジスタ４８に関して競合がなければ、ＦＸＢレジスタ４３はＦＸＡレジスタ４１からセットされる。Ｙレジスタ４９に関して競合がなければ、ＦＹレジスタ４４は初期設定回路２５又はＦＹＳレジスタ４２によりセットされる。競合は次のような形をとり得る。

Ｍ１有効、又は

Ｍ２有効且つＰレジスタ競合、又は

ＸＡが前の乗算に対し既に有効で且つＸＢ競合

ＦＰレジスタ４５は、Ｍ２が有効で且つＰレジスタ５２についての競合がないときに、ＦＸＢレジスタ４３によりセットされる。ＦＰレジスタ４５の有効結果ビット（ＶＲ）は、ＦＰＲアレイ２８への他の書込みに依存しているので、別に維持される。かくして、内部ハザードに遭遇するまで、又は乗算パイプ中の命令の完了を妨げる外部ハザードに遭遇することなくＰレジスタ５２が有効になるまで、パイプ中をできるだけ遠くまでデータを流す内部パイプ制御により乗算パイプが維持される。

ＲＸタイプのロード命令に用いられるパイプ３の構成を第７図に示す。パイプ３は２サイクルしか要せず、その間データは単にパイプ中を流れるだけであつて、その他の処理は行われない。命令はデコーダ２２で解読され、ＦＰＵはＤバス・スタック制御回路３０からのＤバス有効信号を待つ。サイクル１では、データがＤバス・スタック２９のデータ・レジスタ（ＤＲＥＧ）９１に受取られる。このデータは、サイクル２の間にＦＰＲアレイ２８へ送られる。これを監視する制御部は、ＤＲＥＧ有効レジスタ７１を含むＤバス・スタック制御回路３０である。

上述以外の機能（雑機能）を遂行するパイプ４の構成を第８図に示す。このパイプは、Ｄバス・スタック２９の一部を成すＤＲＥＧ９１の出力に接続された間接アドレス・レジスタ６１、同じくＤＲＥＧ９１の出力に接続された状況ワード・レジスタ６２、Ｃバス・レジスタに接続された再試行状況レジスタ６３、及びＣバス・レジスタに接続された間接アドレス・モード・レジスタ６４を含む。状況ワード・レジスタ６２及び再試行状況レジスタ６３の出力はＤＲＥＧ９１及びＦＰＲアレイ２８に接続される。データは、Ｄバスから間

接アドレス・レジスタ６１及び状況ワード・レジスタ６２にロードされる。間接アドレス・レジスタ６１はマイクロコード・モードで使用され、ＦＰＲアドレスを含む。状況ワード・レジスタ６２はＦＰＵの状況、例えば例外や検査機構の状況を維持する。

第８図のパイプ４で遂行される維機能の命令は５つの基本グループに分けられる。

1. 読取り（簡単なビット操作を伴なうことがある）に続いてＦＰＲアレイ２８への書込みを基本的に１サイクルで行うロードＲＲグループ。

2. ＦＰＲアレイ２８以外の間接アドレス・レジスタ６１や状況ワード・レジスタ６２へのロードを行うロードＲＸグループ。

3. この第３グループの命令は２サイクルで実行される。サイクル１では、ＤＲＥＧ９１をロードする一方、ＤＲＥＧ有効レジスタ７１が有効になるのを待ち、サイクル２では、ＤＲＥＧ９１からＤＲＥＧ有効レジスタ７１へのロードを行う。Ｃバスからの情報でレジスタをセットする命令もこのグループに含まれる。例えば、再試行情報を維持する再試行状況レジスタ６３、及びアドレス指定のためのモード・ビットを維持する間接アドレス・モード・レジスタ６４は、このような命令によりセットされる。

4. サイクル１でＦＰＲアレイ２８の読取りを行ってＤＲＥＧ９１に書込み、サイクル２でデータをＤバスへ出力してＤバス有効を知らせる記憶ＲＸグループ。

5. 最後の第５グループは、ＦＰＲアレイ２８以外のレジスタの内容を２サイクルでＤバスに置く記憶タイプの命令を含む。第５グループの命令で読取られる２つのレジスタは、状況ワード・レジスタ６２及び再試行状況レジスタ６３である。

Ｄバス・スタック制御回路３０のレジスタを除くと、パイプ４の動作に関連する制御レジスタはない。

Ｄバス・スタック２９及びＤバス・スタック制御回路３０の構成を第９ａ図及び第９ｂ図に示す。

Ｄバス・スタック制御回路３０は、ＤＲＥＧ有効レジスタ７１、ＤＣバス・レジスタ７１、デコーダ７３、ＤＳ５レジスタ７４、ＤＳ４レジスタ７５、ＤＳ３レジスタ７６、ＤＳ２レジスタ７７、ＤＳ１レジスタ７８、Ｓ２有効レジスタ７９、Ｓ１有効レジスタ８０、及びＭＵＸ選択回路８１を含む。Ｄバス・スタック２９は、Ｄバスに接続されたＳ２レジスタ９３、Ｓ２レジスタ９３の出力及びＤバスに接続されたＳ１レジスタ９２、並びにＳ１レジスタ９２の出力及びＤバスに接続されたＤＲＥＧレジスタ９１を含む。

ＤＳ１からＤＳ５までのレジスタ７８～７４は次のようなビットを含んでいる（第９ｂ図参照）。

ビット０～２（ＩＡＤＤＲ）－命令アドレス。３ビットの命令アドレスがゼロでなければ、境界間ロードが生じることを示す。境界間ロードは２つのＤバス有効信号（データの各部につき１つ）を必要とする。

ビット３及び４（ＰＩＰＥ　ＮＯ）－データを送るべきパイプを一意的に識別するパイプ番号。

ビット５（ＶＩ）－有効命令ビット。

ビット６（ＤＲ）－当該命令のためのデータがＤバス・スタック２９のＤＲＥＧ９１にあることを示す。

ビット７（Ｓ１）－当該命令のためのデータがＤバス・スタック２９のＳ１レジスタ９２にあることを示す。

ビット８（Ｓ２）－当該命令のためのデータがＤバス・スタック２９のＳ２レジスタ９３にあることを示す。

ビット９（ＥＸＥ）－命令が実行中で、適切なパイプが初期設定されたことを示す。

データを受取つた時、そのデータを前述のようにしてパイプ中を流す適切な命令はどれかを見出さなければならない。これは混沌状態を引き起こす。次に、この混沌状態について詳述する。

通常のＲＲタイプのオペレーションにおいては、インターロックが存在しない限り、データはＦＰＲアレイ２８から来る。その場合、ローカル・パイプ制御部は、すべてのパイプの状況レジスタの

アドレス・フィールドを比較することにより適切なデータを見つける。ＲＸタイプのオペレーションの場合はもう少し難しい。オペランド１は同じようにして見つけられるが、オペランド２はデータ・バスから来る。データ・キャッシュを含む他のバス・ユニットは、ＦＰＵが命令を受取るのと同時にデータを供給するよう要求され、その時そのバス・ユニットが使用中であれば、ＦＰＵによる命令及び関連データの受取りが同期的に行われず、その結果として、データのアンダーフロー又はオーバーフローが生じ得る。Ｄバス・スタック制御回路３０は、ＤＲＥＧ９１が適切なパイプに対して有効になると、その旨をローカル・パイプライン機構２６に知らせる。これは、データを分離する際に、特に幾つかのパイプでデータが不足している状態でデータ・バスが有効になつた時、極めて重要である。従つて、Ｄバス・スタック制御回路３０はデータと命令を順序づける。これが終ると、パイプライン機構２６はパイプを通してデータを移動させ、動的活動記録テーブル２７が

れた時に生じる。メモリは読出したデータを直ちに適切なパイプへ入れることはできないので、データをＦＰＵへ送る。従つて、データをスタックすること、及び最終的にデータを送るべきパイプのパイプ番号でデータを識別することが必要である。

ハンドシエーク／大域ハザード回路２３の構成を第１０図に示す。この回路２３は、パイプ１新命令競合回路１０１、パイプ２新命令競合回路１０２、パイプ３新命令競合回路１０３、パイプ４新命令競合回路１０４、大域競合回路１０５、ハンドシエーク組合せ論理回路１０６、及び割込みハンドシエーク組合せ論理回路１０７を含む。回路１０１～１０４はすべてパイプライン機構２６及びデコーダ２２（第１図参照）に接続される。大域競合回路１０５は動的活動記録テーブル２７に接続され、他のユニットからのハンドシエーク信号に応答して、ＦＰＵ使用中信号をＣＰＵへ送つたり、ユニット使用中信号を初期設定回路２５へ送つたりする。ハンドシエーク組合せ論理回路

命令の完了を認めるのを待つ（あとで述べるように、命令の完了は、動的活動記録テーブル２７において最も古い命令、すなわち一番下のエントリのパイプ番号と当該パイプのパイプ番号とを比較することにより検出できる）。このようにして、データの流れが始めから終りまで制御される。更に、メモリはデータが何時要求されたかには無関係に、可能になつた時点でデータを供給するため、データのオーバーフロー又はアンダーフローが生じる可能性があり、この点でもＤバス・スタック制御回路３０が必要である。アンダーフローは、幾つかのパイプが初期設定されているにもかかわらずメモリからのデータ供給が遅い時に生じる（メモリからのオペランドを必要としないＲＸタイプの場合を除く）。複数のパイプがデータを待つている状態でデータが到着した時には、そのデータをどのパイプに送るかを正しく選択する必要がある。オーバーフローは、１以上のパイプが一杯であつて、メモリからのデータを使用する複数の命令がＣＰＵからＦＰＵ及びメモリへ同時に送ら

１０６は大域競合回路１０５に接続され、他のユニットからのハンドシエーク信号に応答して、ＦＰＵ　ＡＣＫ信号をＣＰＵへ送つたり、有効命令信号を初期設定回路２５へ送つたりする。割込みハンドシエーク組合せ論理回路１０７はハンドシエーク組合せ論理回路１０６及び例外処理回路１９に接続され、ＦＰＵ割込み信号をＣＰＵへ送る。

第１０図の回路２３へ入力を供給するのは、他のユニット（ＰＢＵ）、パイプライン機構２６、動的活動記録テーブル２７、及びデコーダ２２である。デコーダ２２からの入力は、新しい命令がＣバス上にあるかどうか、及びどのパイプで新しい命令を実行しなければならないかを回路２３に知らせる。パイプライン機構２６からの入力は、各パイプについてその第１ステージで何らかのハザードが生じているかどうかを示す。他のユニットからの入力（ハンドシエーク信号）は、ＦＰＵの外部にハザードが存在するかどうかをみるために回路２３により監視される。動的活動記録テーブル２７からの入力は、何らかの内部大域ハザー

ドが生じているかどうかを示す。ハザードがなく、新しい命令を実行すべきパイプで競合がなければ、ＦＰＵ　ＡＣＫ信号がハンドシエーク組合せ論理回路１０６からＣＰＵ及び他のユニットへ送られる（命令が２以上のバス・ユニットに対するものであつて、ＡＣＫが抑止される場合を除く）。新しい命令があつても、ハザード又は競合が存在していると、ユニット使用中信号が初期設定回路２５へ送られ、ＦＰＵ使用中信号がＣＰＵ及び他のＰＢＵへ送られる。更に、新しい命令は命令スタック２１のＣバス・レジスタ２１ｂに保持される。この命令は、ＡＣＫを出せるサイクルが生じるまで、後続のサイクルの間新しい命令とみなされる。初期設定回路２５へ送られる有効命令信号は、Ｃバス・レジスタ２１ｂにある命令が適正ＣＰＵハンドシエークを交していてＦＰＵで実行すべきであることを示す早期信号である。これはハザード及び競合には依存しない。回路２３から送られる最後の信号は、例外条件を示すＦＰＵ割込み信号であり、ＣＰＵへ送られる。例外条件は前の命令に対する他のＰＢＵの応答に依存するので、ＦＰＵ割込み信号は例外条件を調べる例外処理回路１９からゲートされねばならない。パイプライン式の計算機においては、複数の命令が出されて同時に実行されるため、同じサイクルで複数の例外が生じる可能性がある。命令セットはＳＩＳＤであるから、命令が順次に実行されているかの如くに例外を処理しなければならない。これは、ハンドシエーク制御から決定される条件でＦＰＵ割込み信号をゲートすることにより為される。このようにして、ハンドシエーク／大域ハザード回路２３によるハンドシエーキング及び大域ハザード検出が達成される。

初期設定回路２５の構成を第１１図に示す。初期設定回路２５は、ハンドシエーク／大域ハザード回路２３、デコーダ２２、及びＭＩＭＤ／ＳＩＳＤ切替え回路２４からの出力に応答する。ハンドシエーク／大域ハザード回路２３からの出力は有効命令信号及びＡＣＫ信号を含む。デコーダ２２からの出力は次のような情報を含む：命令のデータの長さに関係する長さ情報（短いデータはゼロ充填が必要である）；ＦＰＲアレイ２８への書込みを行う時にアドレス情報を供給する書込みタイプ情報；ＦＰＲアレイ２８からデータを取出す時のアドレス情報に関係するＦＰＲアドレス情報；初期設定回路２５へ入力される命令がＲＸタイプかどうかを示すＲＸ命令情報（ＲＸタイプの命令では、命令のデータの２番目の部分は、ＦＰＲアレイ２８ではなくて主記憶装置から供給される）；パイプ１～４の制御に必要なその他の情報；及びパイプ１～４のうちの１つを識別するパイプ番号。デコーダ２２がパイプ番号を供給する理由は、各パイプが特定の命令に関連して機能するように特殊化されており、どのタイプの命令が初期設定回路２５へ入力されるか、従つてどのパイプに入力命令を入れるべきかをデコーダ２２が知つているからである。これは、命令解読というデコーダ２２の機能から考えて当然である。ＭＩＭＤ／ＳＩＳＤ切替え回路２４の出力は、ＭＩＭＤ／ＳＩＳＤ切替え回路２４の切替え位置に関係する情報を与える。

初期設定回路２５は、それぞれ同じ選択信号及び同じ情報信号を受取る５個のマルチプレクサ（ＭＵＸ）１１１～１１５を含む。各マルチプレクサは、選択信号によつて選択されると、情報信号を通過させる。選択信号は、(1)ハンドシエーク／大域ハザード回路２３からの有効命令／ＡＣＫ信号出力、及び(2)デコーダ２２からのパイプ番号出力により構成される。情報信号は、(1)長さ、(2)書込みタイプ、(3)ＦＰＲアドレス、(4)ＲＸ、(5)ＭＩＭＤ／ＳＩＳＤ、及び(6)その他の情報を含む。マルチプレクサ１１１～１１５のうちの１つがパイプ番号信号及び有効命令／ＡＣＫ信号を含む選択信号により選択されると、長さ情報、書込みタイプ情報、ＦＰＲアドレス情報、ＲＸ命令情報、ＭＩＭＤ／ＳＩＳＤ切替え情報、及びその他の情報を含む情報信号がパイプライン回路２６ａ～２６ｄのうちの１つ又は動的活動記録テーブル２７へ送られる。

次に、第１図～第８図を参照しながら、本発明

に従う動的ＭＩＭＤパイプラインの動作について述べる。

第１図において、それぞれ複数の命令を含む複数の命令ストリームが、パイプライン機構２６での実行を待つているものとする。これらの命令ストリームの中から第１図の動的ＭＩＭＤパイプライン（ＦＰＵ）２０へ入力される複数の命令が選択回路（図示せず）により選択され、１つずつＣバスを介して動的ＭＩＭＤパイプライン２０へ入力される。命令は命令スタック２１に受取られ、デコーダ２２で解読される。デコーダ２２は、受取つた命令がＦＰＵで実行できるものであるかどうかを調べる。命令の受取りはハンドシエーク／大域ハザード回路２３により確認され（ＡＣＫ信号がＣＰＵへ送られる）、命令はできるだけ早い機会に実行される。ＣＰＵは、ＦＰＵの連続処理が可能であることを前提にして、命令を連続的に第１図のＦＰＵへ送る。

データ及び１以上の命令がＦＰＵへ入力できる状態にあるものとすると、ＦＰＵは、デコーダ２２で命令を１つずつ解読し、パイプラインがＳＩＳＤモードで動作するのかＭＩＭＤモードで動作するのかをＭＩＭＤ／ＳＩＳＤ切替え回路２４で判断し、何らかのハザードが存在しているかどうかをハンドシエーク／大域ハザード回路２３で調べ、どのパイプが命令を実行するのかをデコーダ２２及び初期設定回路２５で決定し、適切なパイプが何時完了してその中に入力命令を入れられるかを動的活動記録テーブル２７で判断し、これらのことがすべて終ると、ＦＰＵは命令を最大速度で一時に１つずつパイプライン機構２６へ入れることによつて命令を実行しようとする。初期設定回路２５及びハンドシエーク／大域ハザード回路２３は、命令ストリームが始まろうとしているのか終ろうとしているのかを判断し、命令ストリーム及び単一命令の正常終了又は異常終了をＣＰＵに知らせる。ハザード存在の可能性があるため、初期設定回路２５及びハンドシエーク／大域ハザード回路２３は、パイプ１～４のうちの１つがいつ利用されるのか、及び命令がいつ必要になるのかを判断する。前述のように、各パイプは特定カテゴリの命令を処理するように特殊化されており、従つて新しい入力命令がデコーダで解読されて、命令のタイプが決定されると、使用する特定のパイプを識別できる。初期設定回路２５はデコーダ２２からパイプ番号を受取り、マルチプレクサ１１１～１１５を介して対応するパイプへ新しい命令を送る。命令がパイプ１～４のうちの１つに入ると、そのパイプの識別情報を記録すべく動的活動記録テーブル２７が更新される。パイプ１～４はフイードバック形式で動的活動記録テーブル２７に接続されているので、パイプ中での特定の命令の実行状況は継続的に動的活動記録テーブル２７に記録される。従つて、デコーダ２２及び初期設定回路２５で、別の新しい命令が特定のパイプに対応していると識別され、そのパイプに入る準備ができると、そのパイプに内部ハザード（例えば、データ・インターロック又はパイプ満杯）が存在しているかどうかを調べるため、該パイプの最初のステージに関連するパイプ制御部に記憶されている情報が読取られる。内部ハザードが存在していると、新しい命令はこのパイプに入れない。その時、この特定のパイプのパイプ番号が動的活動記録テーブル２７の一番下のエントリのパイプ番号欄（ＰＩＰＥ　ＮＯ）にあるかどうかが調べられる。例えば、特定のパイプが番号Ｘにより識別され、３つのパイプライン・ステージを有していて、動的活動記録テーブル２７がパイプ番号欄にパイプ番号Ｘを含む３つのエントリを有していると、パイプＸは命令で一杯であり、従つて新しい命令を入れるためには、パイプＸ中の最も古い命令を完了させなければならない。テーブル２７における最も古い（一番下の）エントリのパイプ番号欄がＸになつていると、パイプＸ中の最も古い命令を完了させることができ、これにより、新しい命令をパイプＸに入れられるようになる。要約すると、テーブル２７の働きは、パイプ１～４の制御部と一緒になつて、すべてのパイプが適正に利用され且つ命令が始めから終りまで適正に実行されるように、継続的な補助を与えることにあ

る。各パイプは特定カテゴリの命令を実行し、その外部的な制御は動的活動記録テーブル２７及び初期設定回路２５が受持ち、内部的な制御は内部パイプ制御部が受持つ。各パイプは、複数の命令ストリームに属する複数の命令を実行することもできる。

次に、第１図及び第４図を参照しながら、動的活動記録テーブル２７の動作について説明する。

ＡＣＫ信号がハンドシエーク／大域ハザード回路２３から送られるか、又は使用中信号が回路２３によつて落とされると、入力命令に関する情報が動的活動記録テーブル２７の１つのエントリに入れられる。第４図の例では、テーブル２７は８個のエントリ（１～８）を持つている。このテーブル２７は、入力命令をどのように完了させるかを決定するのに必要な幾つかのパラメータを含む。キー・パラメータは、当該命令を実行するパイプを示すパイプ番号である。アーキテクチヤ上の制約から順次的な実行に見せかける必要があるため、動的活動記録テーブル２７はサイクル毎に読出され、入力命令の実行を次に完了するのはどのパイプかを示す。各サイクルでテーブル２７から読出された情報は、選択されたパイプが完了を待つているかどうかを見るために、そのパイプの内部状況（状況レジスタにある）と比較される。もし完了を待つていれば、命令は完了され、動的活動記録テーブル２７中の対応するエントリが消去される。云い換えれば、番号Ｘのパイプが命令を実行中で、パイプ番号Ｘが動的活動記録テーブル２７の最も古い（一番下の）エントリのパイプ番号欄に記録されている場合、パイプＸにおける最も古い命令の完了は、テーブル２７中の当該命令に関連するパラメータとパイプＸの状況レジスタの内容とを比較することを含む。パイプＸの状況レジスタが最も古い命令の実行完了を示していると、その実行結果はパイプＸから適切な宛先の方へ送り出される。パイプＸの内容は１ステージだけシフトされ、新しい命令を実行のために挿入できるようにする。

次に第９図を参照しながら、Ｄバス・スタック２９及びＤバス・スタック制御回路の動作について説明する。

第９ａ図において、Ｄバス・スタック制御回路３０のレジスタ７４～７８は命令用のスタックとして働き、Ｄバス・スタック２９のレジスタ９１～９３はデータ用のスタックとして働く。Ｃバス上の命令はＤＣバス・レジスタ７２及びデコーダ７３の働きによつてスタックされる。ＤＣバス・レジスタ７２は、使用中信号とは無関係に、サイクル毎にＣバスをラツチする。デコーダ７３は、前の使用中状況の認識に基いて、いつ新しいＤバス・ロード命令がＦＰＵに入るかを示す。かくして、データが到着して命令の実行が始まるまで、Ｄバス・ロード・タイプの各命令に関する情報が維持される。他の重要な状況レジスタは、ＤＲＥＧ有効レジスタ７１、Ｓ１有効レジスタ８０及びＳ２有効レジスタ７９である。これらは何れも２ビットのレジスタであつて、データが部分的に有効か（Ｐ）、完全に有効か（Ｆ）を示す。レジスタ７４～７８にある命令が実行されておらず且つレジスタ９１～９３にデータが入つていると、データのオーバーフローが生じる。一方、幾つかの命令がレジスタ７４～７８に入つているが、レジスタ９１～９３に十分なデータがなければ、データのアンダーフローが生じる。入力データはＤバス・スタック２９のレジスタ９１～９３にスタックされ、入力命令はＤバス・スタック制御回路３０のレジスタ７４～７８にスタックされるが、Ｄバス・データを必要とするスタックされた命令とスタックされたデータとの間には１対１の対応関係がある。Ｄバスからの一組のデータをパイプライン機構２６へ送る場合には、パイプ１～４のうちのどのパイプがこの一組のデータを受取るのかを決定する必要がある。Ｄバス・スタック２９のレジスタ９１～９３にスタックされたデータと、Ｄバス・スタック制御回路３０のレジスタ７８～７６にスタックされた命令との間には１対１の対応関係があるので、到着した一組のデータは、Ｄバス・スタック制御レジスタ７４～７８のうちの一番下のレジスタ７８のビット３及び４（ＰＩＰ

Ｅ　ＮＯ）によつて表わされるパイプ番号を持つたパイプ（パイプ１～４のうちの１つ）に送られる。

最後に、簡単な命令ストリームの例を示す第１２図を参照しながら、この命令ストリームが第１図のハードウエア中をどのように流れるかを説明する。

第１２図の命令ストリームは次の３つの命令から成つている。

(1) ＦＰＲ１へのロードを行うＲＸタイプのロード命令。

(2) ＦＰＲ１及びＦＰＲ２の長精度乗算を行つて、長精度結果をＦＰＲ１に書込むＲＲタイプの長精度乗算命令。

(3) ＦＰＲ３及びＦＰＲ４の長精度加算を行つて、長精度結果をＦＰＲ３に書込むＲＲタイプの長精度加算命令。

この命令ストリームから次のようなことがわかる。

(イ) 各命令はそれぞれ異なつたパイプで実行する必要がある。

(ロ) ロード命令は、乗算命令で使用されるレジスタ（ＦＰＲ１）への書込みを行う。この競合の結果、もし書込み（ロード）が生じる前に乗算命令が受取られると、インターロックが生じ得る。

(ハ) 加算命令は、前のロード命令及び乗算命令で使用されるＦＰＲを使用しない。

第１図のハードウエアがこの命令ストリームを各サイクルでどのように処理するかを以下に述べる。

サイクル０－ＣＰＵがＦＰＵに向けて、ＲＸタイプのロード命令をＣバスへ送り出し、Ｃバス・レジスタ２１ｂ（第３図）及びＤＣバス・レジスタ７２（第９図）がこれをラッチする。

サイクル１－ロード命令がデコーダ２２及びデコーダ７３（第９図）で解読される。ハンドシエーク／大域ハザード回路２３が、パイプ３の内部パイプ制御部及び動的活動記録テーブル２７を検査することによつて、ロード命令の開始に問題がないこと、及び大域ハザードが存在しないことを決定する。この結果、ＡＣＫ信号（ＦＰＵ　ＡＣＫ）がハンドシエーク／大域ハザード回路２３からパイプ３を初期設定する初期設定回路２５へ送られる（複数バス・ユニットＰＢＯの場合はデータ・キヤツシユが両ユニットに対して応答するので、ＦＰＵ　ＡＣＫは他のユニットには送られない）。初期設定回路２５は、ロード命令を動的活動記録テーブル（ＤＨＴ）２７に置くための情報も与える。ロード命令がＲＸタイプであることがデコーダ７３（第９図）で検出されているので、Ｄバス・スタック制御回路３０ではロード命令の情報がＤＳ１レジスタ７８に置かれる。また、データ有効信号がＤバスからＤＲＥＧ９１に受取られているため、ＤＲＥＧ有効レジスタ７１が活動状態になる。このサイクルの間、ＣＰＵはＦＰＵに向けて、ＲＲタイプの乗算命令をＣバスへ送り出し、Ｃバス・レジスタ２１ｂ及びＤＣバス・レジスタ７２がこれをラッチする。

サイクル２－ＲＲタイプの乗算命令がデコーダ２２及び７３で解読される。ハンドシエーク／大域ハザード回路２３が、パイプ２の内部パイプ制御部及び動的活動記録テーブル２７を検査することによつて、乗算の開始に問題がないこと、及び大域ハザードが存在しないことを決定する。この結果、ＡＣＫ信号（ＦＰＵ　ＡＣＫ）がＣＰＵ及び他のＰＢＵへ送られ、初期設定回路２５がパイプ２を初期設定すると共に、ＦＰＲ２をＦＰＲアレイ２８からＦＬＰバスへ読出し、Ｙレジスタ４９にラッチさせる（第６ａ図）。初期設定回路２５は、乗算命令を動的活動記録テーブル２７に置くための情報も与える。更に初期設定回路２５は、ＦＰＲ１がロード命令によりインターロックされ、従つて乗算のオペランド１がインターロックされることを、動的活動記録テーブル２７を解読したハンドシエーク／大域ハザード回路２３から知らせ、デコーダ２２からの情報に基いてＦＸＡレジスタ４１及びＦＹレジスタ４４を初期設定する。Ｄバス・スタック制御回路３０のデコーダ７３（第９図）

では、乗算命令がDバスを使用しないこと、及びDREG91によつてインターロックされることを検出しているので、DREG91がXAレジスタ46（第6a図）にロードされる。DREG91はこのサイクルで有効であり、ロード命令を完了させるためFPRアレイ28がロード（書込み）される（ロード命令の完了は、次に完了するのがパイプ3であることがDREG有効レジスタ71及び動的活動記録テーブル27により示されると可能である）。CPUはFPUに向けて、RRタイプの加算命令をCバスへ送り出し、Cバス・レジスタ21b及びDCバス・レジスタ72がこれをラッチする。

サイクル3－RRタイプの加算命令がデコーダ22及び73で解読される。ハンドシエーク／大域ハザード回路23が、パイプ1の内部パイプ制御部及び動的活動記録テーブル（DHT）27を検査することによつて、加算の開始に問題がないこと、及び大域ハザードが存在しないことを決定する。この結果、ACK信号（FPU ACK）がハンドシエーク／大域ハザード回路23からCPU及び他のPBUへ送られ、初期設定回路25がパイプ1を初期設定すると共に、FPR3及びFPR4をFPRアレイ28からAレジスタ35及びBレジスタ36へ読出す（第5a図）。初期設定回路25は、加算命令を動的活動記録テーブル27に置くための情報も与え、更にインターロックが存在しないことを、動的活動記録テーブル27を解読したハンドシエーク／大域ハザード回路23から知らせる。また初期設定回路25はデコーダからの情報により、FAレジスタ31及びFBレジスタ32を初期設定する。第6a図において、MUX選択回路53で決定された競合のため、乗算ハードウエアはオペランドをYレジスタ49に保持し、3Xハードウエア47で3倍乗算を実行し、XB／3Xレジスタ48にラッチする。乗算命令のための制御情報はFYレジスタ44に保持され、FXAレジスタ41からFXBレジスタ43へ転送される。第5a図において、整列レジスタ34による整列（アラインメント）もこのサイクルで行われる。

サイクル4－第6a図において、両方のオペランドがXB／3Xレジスタ48及びYレジスタ49で使用可能であり、かくてM1サイクルが開始し、乗算を行うためにM1ハードウエア50が使用される。第5a図において、加算のサイクル2に入り、加算器37が動作して、その出力がSレジスタ38にラッチされる。FAレジスタ31のデータがFSレジスタ33へ転送される。

サイクル5－第6a図において、乗算のM2サイクルが開始してM2ハードウエア51が使用され、その結果がPレジスタ52にラッチされる。FXBレジスタ43がFPレジスタ45をセットする。動的活動記録テーブル（DHT）27は、乗算命令のためのパイプ2が次に完了しなければならないことを示しているので、第5a図のパイプ1は、Sレジスタ38のステージで待機していなければならない。

サイクル6－第6a図において、Pレジスタ52が有効であることがFPレジスタ45により示され、且つこのパイプが次に完了することを動的活動記録テーブル（DHT）27が示しているので、乗算命令が完了する。乗算を完了させるため、動的活動記録テーブル27は書込みアドレス及び長さ情報をFPRアレイ28へ供給する。加算はまだSレジスタ38のステージで待機している。

サイクル7－Sレジスタが有効であることがFSレジスタにより示され、且つ加算パイプが次に完了することを動的活動記録テーブル（DHT）27が示しているので、加算が完了する。

これで、第12図の命令ストリームの実行が完了したことになる。

4.図面の簡単な説明

第1図は本発明に従う動的MIMDパイプラインの構成を示すブロック図。

第2図は従来の簡単的なMIMDパイプラインの概略を示すブロック図。

第３図は命令スタック２１の構成を示すブロック図。

第４図は動的活動記録テーブル２７の内容を示す図。

第５ａ図及び第５ｂ図はパイプライン回路２６ａの構成を示すブロック図。

第６ａ図及び第６ｂ図はパイプライン回路２６ｂの構成を示すブロック図。

第７図はパイプライン回路２６ｃの構成を示すブロック図。

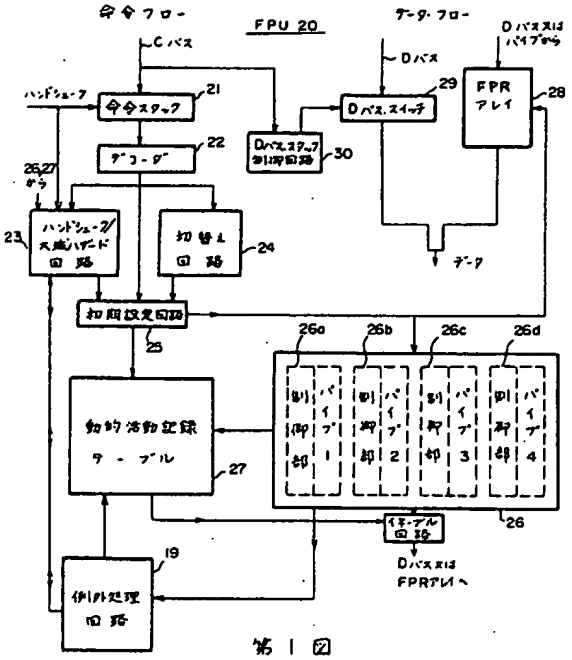
第８図はパイプライン回路２６ｄの構成を示すブロック図。

第９ａ図及び第９ｂ図はＤバス・スタック２９及びＤバス・スタック制御部３０の構成を示すブロック図。

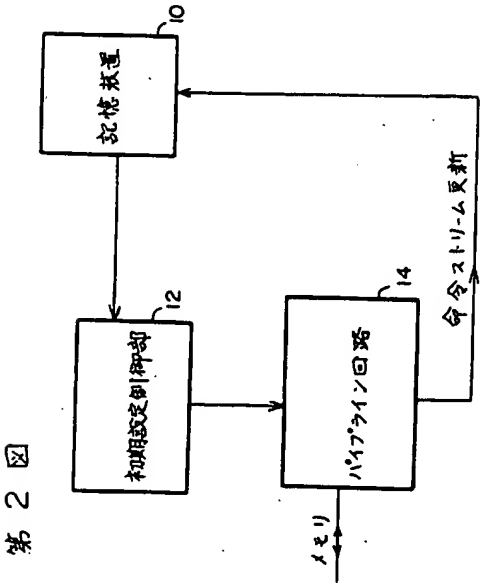
第１０図はハンドシェーク／大域ハザード回路２３の構成を示すブロック図。

第１１図は初期設定回路２５の構成を示すブロック図。

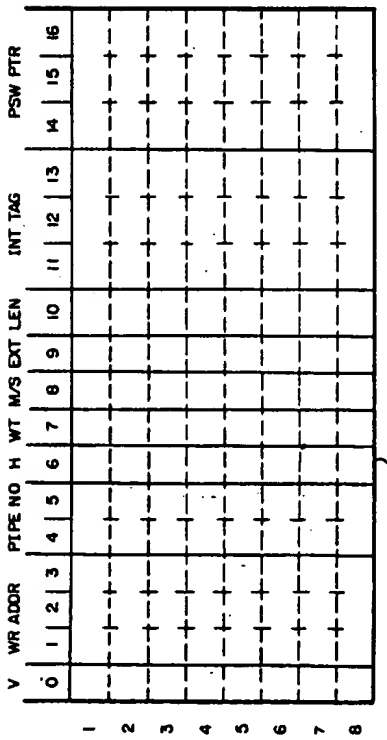
第１２図は命令ストリームの一例を示す図。



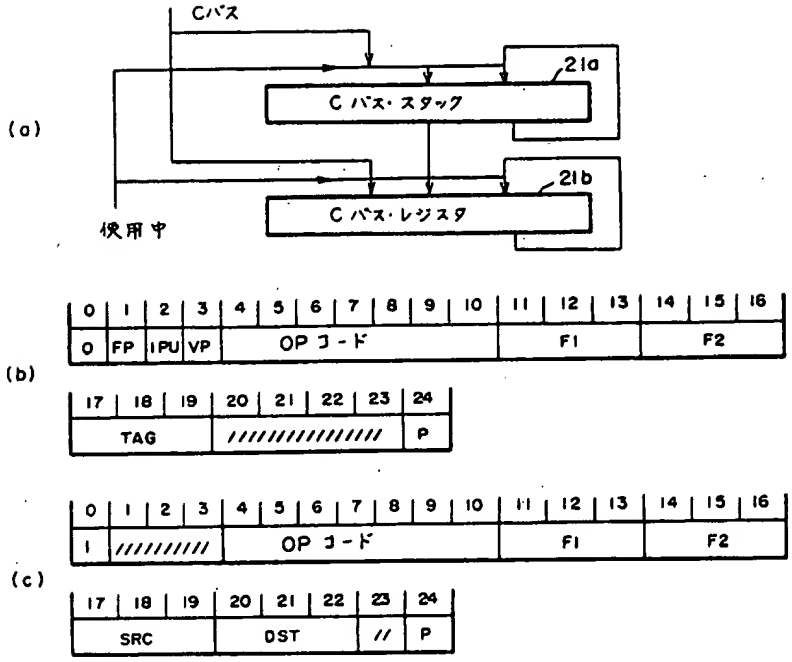
第１図



第２図

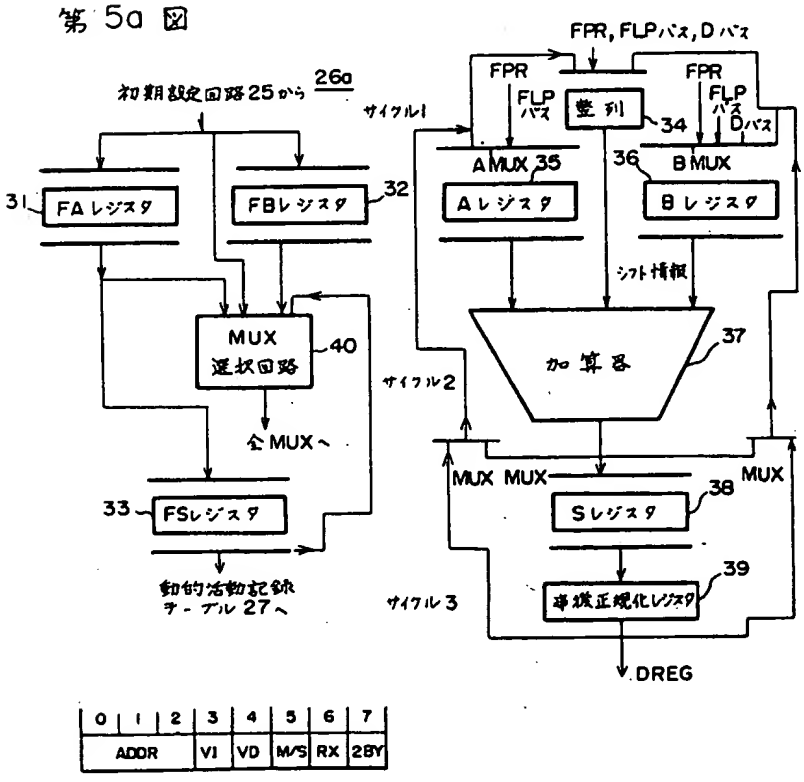


第４図



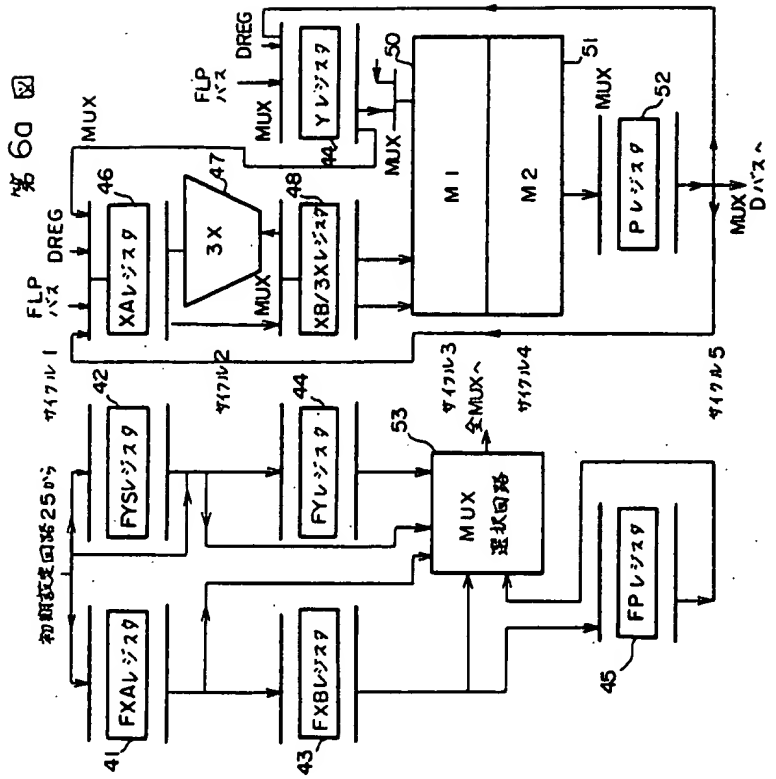
第3図

第5a図

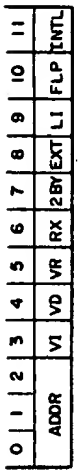


第5b図

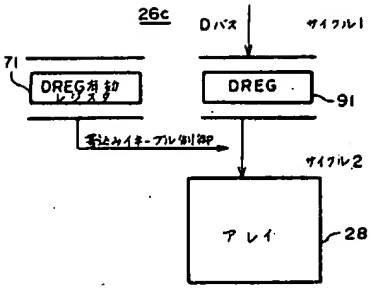
第6a図



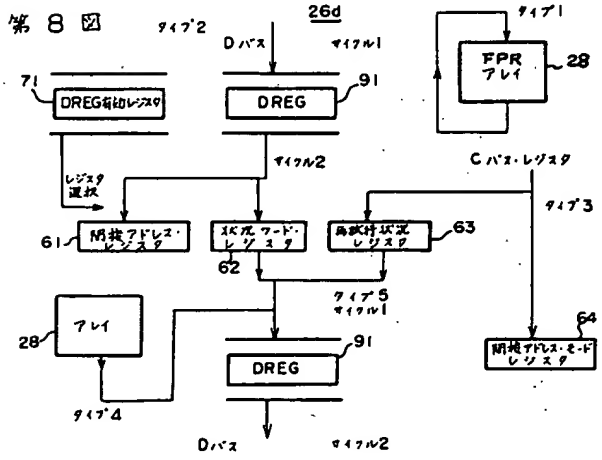
第6b図



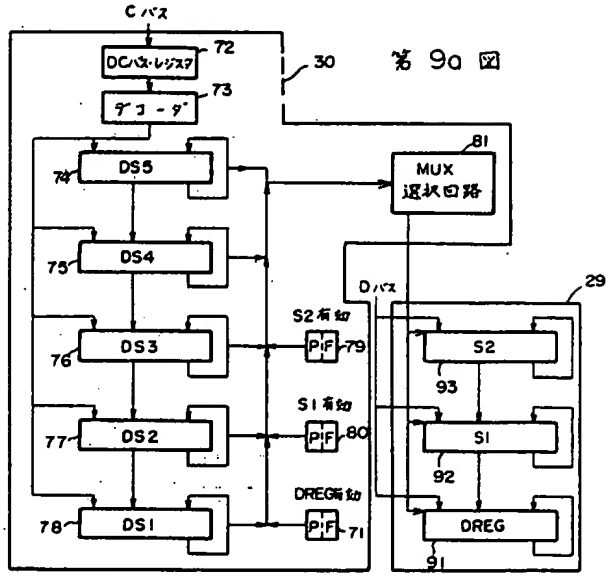
第7図



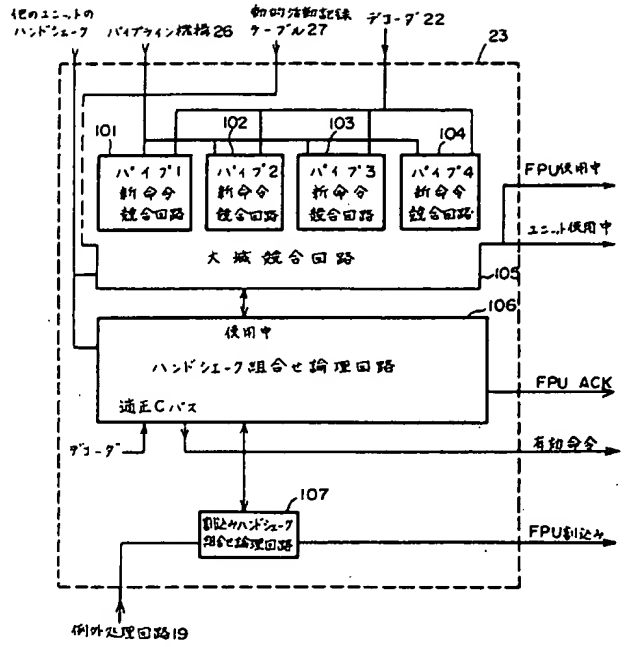
第8図



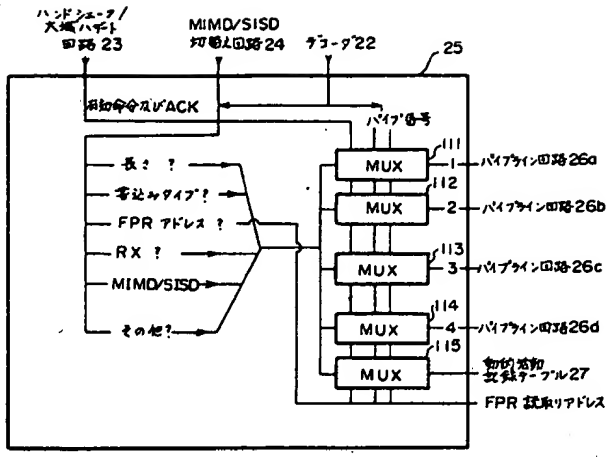
第9a図



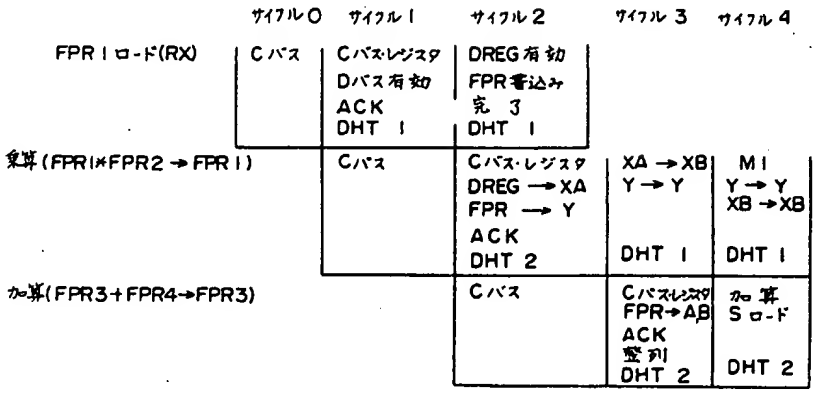
第9b図



第10図



第11図



第12図